CHAPTER THREE

3.1 Methodology

3.1.1 Introduction

This chapter outlines the methodology adopted in the development of the Stock Price Prediction Web Application. The methodology encompasses the chosen software development model, justification for its use, a breakdown of the activities involved in each phase, and tools used throughout the development process.

3.1.2 Chosen Software Development Process Model: Agile Methodology

3.1.3 Justification for Choosing Agile Methodology

The Agile development methodology was selected due to the iterative and flexible nature of the project. The stock prediction system involves several components including data retrieval, machine learning-based prediction, and a web interface all of which benefit from regular feedback, testing, and improvement. Agile facilitates rapid prototyping, frequent evaluation of working software, and adaptive planning which are ideal for this project.

3.1.4 Outline of Agile Process Model

Agile software development is an iterative and incremental approach to software delivery. It emphasizes collaboration, flexibility, and customer feedback. The core phases of Agile used in this project are:

1. Requirement Gathering & Planning
2. Design
3. Implementation
4. Testing
5. Deployment
6. Evaluation & Feedback

3.1.5 Activities Embarked on in Each Phase

- Requirement Gathering & Planning:
  - Identified the target users (investors, finance students, data analysts).
  - Defined project goals and scope: forecasting stock prices using machine learning.
  - Selected companies and features (date selection, stock symbol, and visualization).

- Chose tools and technologies (Python, Streamlit, Prophet, yfinance, Plotly).

➢ Design:
- Designed wireframes for the user interface.
- Modeled system architecture and interactions using UML diagrams.
- Defined user input/output flow and interaction design.

➢ Implementation:
- Implemented backend logic using Python and the Prophet forecasting model.
- Retrieved data using yfinance and processed it into suitable format.
- Built frontend using Streamlit for real-time user interaction.
- Integrated Plotly for data visualization.

➢ Testing:
- Unit tested each component for correctness.
- Validated the predictions generated by the Prophet model.
- Conducted UI/UX testing to ensure usability.

➢ Deployment:
- Hosted the Streamlit application locally and on the cloud for broader access.
- Made use of GitHub for version control and project management.

➢ Evaluation & Feedback:
- Gathered feedback from potential users.
- Iteratively improved the app based on feedback (e.g., simplified date range selection, added more visualizations).

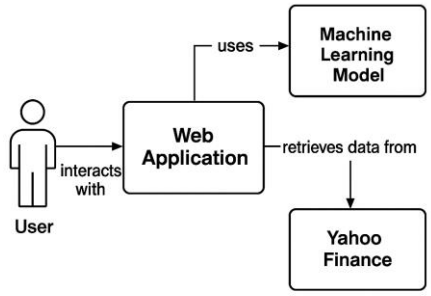
3.2 Proposed System

3.2.1 Introduction of Proposed System

The proposed system is a web-based application designed to allow users to predict and visualize stock prices. The system enables selection of predefined companies, forecasting based on a user-defined date range, and displays historical as well as predicted data in interactive charts.

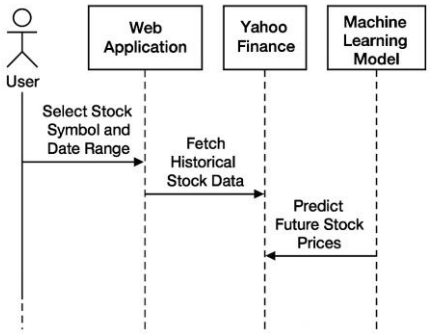
3.2.2 Models of the Proposed System

1. Context Model:



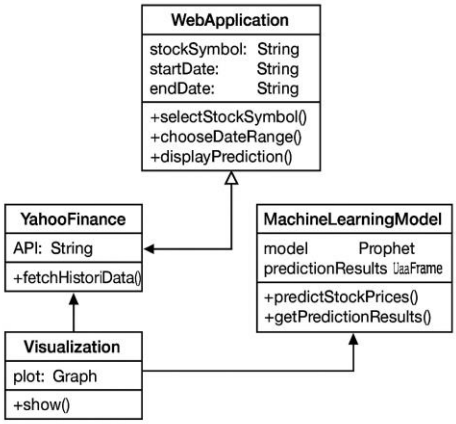
Context diagram

2. Interaction Model (Sequence Diagram):



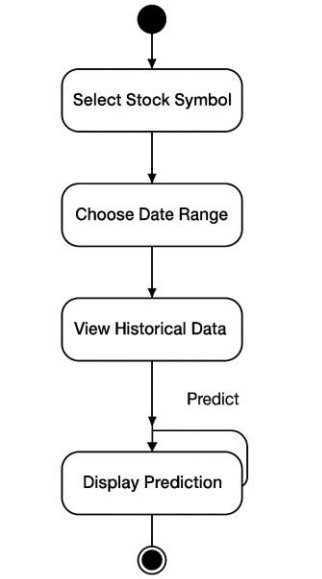
Stock Price Prediction Sequence Diagram

3. Structural Model (Class Diagram):



Structural Model of Stock Price Prediction Web Application

4. Behavioral Model (Activity Diagram):



Behavioral Model of Stock Price Prediction Web Application

3.2.3 New Features of the Proposed System

- Stock selection from a predefined list.
- Customizable date range for forecasting.
- Interactive plots (candlestick, forecast trends).
- Multiple company support.
- Prediction horizon up to 5 years.

3.2.4 Development Tools and Environment

- Development Tools:
    - Python: For logic, data handling, and model integration.
    - Streamlit: For rapid UI development and deployment.
    - Prophet: For time-series forecasting.
    - yfinance: For fetching historical stock data.
    - Plotly: For interactive data visualization.
- Development Environment:
    - IDE: Visual Studio Code
    - Operating System: Windows 10
    - Version Control: Git & GitHub
    - Browser: Google Chrome

3.2.5 Review of the Good Features

- User-friendly interface with real-time interaction.
- Reliable data sourcing using yfinance.
- Accurate forecasts using Prophet.
- Insightful and interactive visualizations with Plotly.
- Scalable design supporting multiple companies and date ranges.

3.2.6 Review of the Bad Features

- Dependency on internet connectivity for data retrieval.
- Limited to only selected companies, limiting its generalizability to other stocks.
- Model accuracy may decrease during highly volatile market conditions.

3.2.7 Summary of the System Review

The Stock Price Prediction Web Application successfully meets its objectives by integrating reliable forecasting models with an intuitive user interface. It provides users with an easy-to-use tool to analyze and predict stock market trends, offering educational and investment benefits. While there are areas for enhancement, especially around model accuracy and offline capabilities, the system as developed is a robust solution for time-series forecasting in finance.